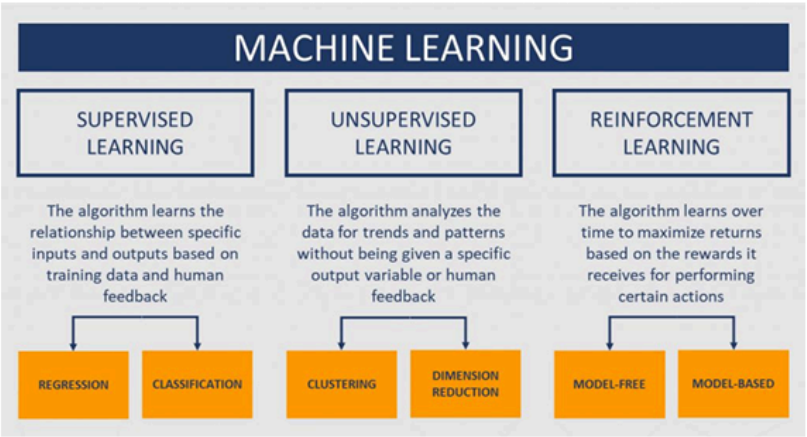
Machine Learning



মেশিন লার্নিংয়ের এই দুটি হলো একদম মূল ভিত্তি। সহজ কথায়, এদের মধ্যে প্রধান পার্থক্য হলো **"শিক্ষক বা লেবেল"** আছে কি নেই। নিচে বিস্তারিত আলোচনা করা হলো:

১. Supervised Learning (তত্ত্বাবধায়নে শেখা)

এখানে মডেলটি একজন শিক্ষকের অধীনে শেখার মতো কাজ করে। আমরা মডেলকে ইনপুট ডেটা (X) এবং তার সাথে সঠিক উত্তর বা **Label (y)** দিয়ে দিই। মডেলটি ইনপুট দেখে আউটপুট মেলাতে শেখে।

- কীভাবে কাজ করে**:** ডাটাতে ইনপুট এবং আউটপুট জোড়ায় জোড়ায় থাকে। যেমন: $[ছবি] \rightarrow \text{বিড়াল}$, $[ছবি] \rightarrow \text{কুকুর}$।
- লক্ষ্য**:** নতুন কোনো অজানা ইনপুট দিলে তার সঠিক আউটপুট প্রেডিক্ট করা।

types (প্রকারভেদ):

Supervised learning প্রধানত দুই প্রকার:

1. **Classification (শ্রেণিবিন্যাস):** যখন আউটপুটটি কোনো ক্যাটাগরি বা গ্রুপ হয়।
    - উদাহরণ*:* ইমেইল কি স্প্যাম নাকি স্প্যাম না? অথবা টিউমার কি ক্যানসার নাকি ক্যানসার নয়?
2. **Regression (প্রতিগমন):** যখন আউটপুটটি কোনো কন্টিনিউয়াস বা সংখ্যাবাচক মান হয়।
    - উদাহরণ*:* আগামীকালের তাপমাত্রা কত হবে? অথবা কোনো বাড়ির দাম কত হতে পারে?

২. Unsupervised Learning (তত্ত্বাবধায়নহীন শেখা)

এখানে কোনো শিক্ষক বা লেবেল থাকে না। আমরা মডেলকে শুধু ইনপুট ডেটা দিই, কিন্তু বলে দিই না যে কোনটা কী। মডেল নিজেই ডেটার মধ্যে মিল, অমিল বা প্যাটার্ন খুঁজে বের করে।

- কীভাবে কাজ করে**:** মডেল ডেটার গঠন (Structure) বিশ্লেষণ করে। যেমন: অনেকগুলো ফলের ছবি দিলে সে লাল গোল ফলগুলোকে একদিকে আর হলুদ লম্বা ফলগুলোকে অন্যদিকে আলাদা করে ফেলবে (যদিও সে জানে না কোনটার নাম আপেল বা কলা)।
- লক্ষ্য**:** ডেটার ভেতরে লুকিয়ে থাকা গ্রুপ বা স্ট্রাকচার খুঁজে বের করা।

types (প্রকারভেদ):

Unsupervised learning প্রধানত তিন প্রকার:

1. **Clustering (গুচ্ছকরণ):** ডেটার বৈশিষ্ট্যের ওপর ভিত্তি করে মিল থাকা ডেটাগুলোকে আলাদা গ্রুপে ভাগ করা।
    - উদাহরণ: কাস্টমারদের কেনাকাটার অভ্যাসের ওপর ভিত্তি করে তাদের বিভিন্ন গ্রুপে ভাগ করা (Customer Segmentation)।
2. **Association (সম্পর্ক নির্ণয়):** একটি জিনিসের সাথে অন্য জিনিসের সম্পর্ক খুঁজে বের করা।
    - উদাহরণ: সুপারশপে যারা ব্রেড কেনে, তাদের দুধ কেনার সম্ভাবনা বেশি থাকে (Market Basket Analysis)।
3. **Dimensionality Reduction:** বিশাল সংখ্যক কলাম থেকে তথ্য না হারিয়ে কলামের সংখ্যা কমিয়ে আনা (যেমন: PCA)।

৩. সহজ তুলনা টেবিল

বৈশিষ্ট্য	**Supervised Learning**	**Unsupervised Learning**
ডেটার ধরন	Labeled Data (ইনপুট + আউটপুট)	Unlabeled Data (শুধু ইনপুট)
শিক্ষক	শিক্ষক থাকে (Labels)	শিক্ষক থাকে না
কাজ	প্রেডিকশন (Prediction)	প্যাটার্ন বা স্ট্রাকচার খোঁজা
অ্যালগরিদম	Linear Regression, SVM, Decision Tree.	K-Means Clustering, PCA, Apriori.

---

৪. বাস্তব জীবনের উদাহরণ

- **Supervised:** একজন ছাত্রকে অনেকগুলো অঙ্ক সমাধানসহ দেওয়া হলো। সে সেগুলো দেখে নিয়ম শিখল এবং পরে নতুন অঙ্ক সমাধান করতে পারল।
- **Unsupervised:** ওই একই ছাত্রকে অনেকগুলো অজানা মুদ্রা দেওয়া হলো। সে জানে না কোন মুদ্রার নাম কী, কিন্তু সে রং আর আকার দেখে সোনা, রুপা আর তামা—এই তিন ভাগে মুদ্রাগুলোকে সাজিয়ে ফেলল।

সারসংক্ষেপ:

যদি তোমার কাছে উত্তর জানা থাকে এবং তুমি ভবিষ্যতে প্রেডিকশন করতে চাও, তবে সেটা **Supervised**। আর যদি তোমার কাছে শুধু বিশাল ডেটা থাকে এবং তুমি জানতে চাও সেখানে কী ধরণের গ্রুপ বা প্যাটার্ন আছে, তবে সেটা **Unsupervised**।

🤖 Reinforcement Learning (RL) কি?

Reinforcement Learning হলো মেশিন লার্নিংয়ের এমন একটি শাখা যেখানে একটি **Agent** (যেমন: একটি রোবট বা গেম প্লেয়ার) একটি **Environment**-এ কাজ করে এবং তার কাজের ফলাফলের ওপর ভিত্তি করে শেখে।

এটি অনেকটা ছোট বাচ্চাকে বড় করার মতো। ভালো কাজ করলে আমরা যেমন তাকে চকলেট দিই (Reward), আর খারাপ কিছু করলে বকা দিই (Penalty)—মেশিনও ঠিক সেভাবেই শেখে।

RL যেভাবে কাজ করে (Main Components):

১. **Agent:** যে শেখে বা কাজ করে (যেমন: একটি রোবট)।

২. **Environment:** যেখানে এজেন্ট কাজ করে (যেমন: একটি মেজ বা গোলকধাঁধা)।

৩. **Action (A):** এজেন্ট যে কাজগুলো করে (যেমন: ডানে বা বামে যাওয়া)।

৪. **State (S):** এজেন্টের বর্তমান অবস্থা।

৫. **Reward (R):** কাজের ফলাফল। ভালো কাজের জন্য $+1$, খারাপের জন্য -1।

প্রক্রিয়া: এজেন্ট একটি অ্যাকশন নেয় $\rightarrow$ এনভায়রনমেন্ট তাকে ফিডব্যাক দেয় $\rightarrow$ এজেন্ট রিওয়ার্ড পায় $\rightarrow$ সে বুঝতে পারে এই কাজটি ভবিষ্যতে আবার করা উচিত কি না।

সুবিধা ও অসুবিধা:

- সুবিধা:
  - যেখানে কোনো আগে থেকে তৈরি ডেটা (Labels) নেই, সেখানে এটি দারুণ কাজ করে।
  - জটিল এবং পরিবর্তনশীল পরিবেশে সিদ্ধান্ত নিতে পারে।
- অসুবিধা:
  - প্রচুর কম্পিউটেশনাল পাওয়ার এবং সময়ের প্রয়োজন।
  - শুরুতে এজেন্ট অনেক ভুল করে (Exploration phase), যা বাস্তব ক্ষেত্রে (যেমন: ড্রাইভলেস কার) বিপজ্জনক হতে পারে।

ব্যবহারের ক্ষেত্র:

- গেম খেলা: যেমন AlphaGo বা দাবা খেলায়।
- রোবোটিক্স: রোবটকে হাঁটতে শেখানো বা জিনিসপত্র ধরতে শেখানো।
- অটোনোমাস ভেহিকল: স্বয়ংক্রিয় গাড়ি চালানো।
- রিসোর্স ম্যানেজমেন্ট: ডাটা সেন্টারের তাপমাত্রা নিয়ন্ত্রণ বা ইনভেন্টরি ম্যানেজমেন্ট।

📉 Overfitting এবং Underfitting

মেশিন লার্নিং মডেল যখন ডেটা থেকে শেখে, তখন এই দুটি সমস্যা সবচাইতে বেশি দেখা যায়।

১. Underfitting (High Bias)

এটি তখন ঘটে যখন মডেলটি এতই সহজ হয় যে সে ডেটার প্যাটার্ন ঠিকমতো বুঝতেই পারে না।

- অবস্থা: ট্রেনিং ডেটাতেও রেজাল্ট খারাপ, নতুন ডেটাতেও রেজাল্ট খারাপ।
- উদাহরণ: তুমি গণিত পরীক্ষার জন্য শুধু সূত্রগুলো মুখস্থ করলে কিন্তু একটাও অঙ্ক প্র্যাকটিস করলে না। ফলে পরীক্ষায় অঙ্ক ঘুরিয়ে দিলে তুমি পারবে না।
- সমাধান: আরও ফিচার যোগ করা বা আরও জটিল অ্যালগরিদম ব্যবহার করা।

২. Overfitting (High Variance)

এটি তখন ঘটে যখন মডেলটি ডেটা শেখার বদলে মুখস্থ করে ফেলে। সে ডেটার ভেতরের 'নয়েজ' বা ভুলগুলোকেও প্যাটার্ন ভেবে নেয়।

- অবস্থা: ট্রেনিং ডেটাতে ১০০% সঠিক রেজাল্ট দেয়, কিন্তু নতুন কোনো ডেটা দিলেই ভুল করে।
- উদাহরণ: তুমি গত ৫ বছরের প্রশ্ন হুবহু মুখস্থ করলে। যদি পরীক্ষায় ওই প্রশ্নগুলোই আসে তবে তুমি ফাটিয়ে দেবে, কিন্তু একটু বাইরে থেকে প্রশ্ন এলেই তুমি ফেল করবে।
- সমাধান: ডেটা বাড়ানো, অপ্রয়োজনীয় ফিচার কমানো (Dimensionality Reduction), বা Regularization ব্যবহার করা।

৩. সহজ তুলনা টেবিল:

বৈশিষ্ট্য	**Underfitting**	**Optimal (Good)**	**Overfitting**
মডেলের জটিলতা	খুব সহজ (Simple)	সঠিক (Balanced)	খুব জটিল (Complex)
ট্রেনিং এরর	অনেক বেশি	কম	একদম নেই
টেস্টিং এরর	অনেক বেশি	কম	অনেক বেশি
উপমা	কিছুই বোঝেনি	বুঝেছে	মুখস্থ করেছে

সারসংক্ষেপ:

Reinforcement Learning হলো অভিজ্ঞতা থেকে শেখা। আর মডেল বানানোর সময় আমাদের লক্ষ্য থাকে এমনভাবে শেখানো যাতে সে **Underfitting** (কিছুই না বোঝা) বা **Overfitting** (সব মুখস্থ করা)—কোনোটাতেই না পড়ে একদম **Balanced** হয়।

অনেক সময় এই দুটোর মধ্যে কনফিউশন তৈরি হয় কারণ দুটোর মধ্যেই কোনো **"**সঠিক উত্তর**" (Label)** আগে থেকে দেওয়া থাকে না। কিন্তু এদের কাজ করার উদ্দেশ্য এবং পরিবেশ সম্পূর্ণ আলাদা।

WHEN TO USE

সহজ পার্থক্যের চাবিকাঠি হলো: **Unsupervised** খোঁজে "প্যাটার্ন" (Data Patterns), আর **Reinforcement** খোঁজে "সেরা উপায়" (Best Strategy)।

নিচে বিস্তারিত বুঝিয়ে দিচ্ছি:

১. **Unsupervised Learning** কখন ব্যবহার করবে**?**

যখন তোমার কাছে অনেক ডেটা আছে কিন্তু তুমি জানো না সেই ডেটাগুলো আসলে কী বোঝাতে চাচ্ছে। এখানে কোনো অ্যাকশন নেওয়ার ব্যাপার নেই, শুধু ডেটাকে চেনা বা বোঝা মূল কাজ।

- উদ্দেশ্য: ডেটার ভেতরে লুকিয়ে থাকা গ্রুপ বা মিল খুঁজে বের করা।
- কখন ব্যবহার করবে:
    - কাস্টমার সেগমেন্টেশন: তোমার কাছে ১ লাখ কাস্টমারের লিস্ট আছে। তুমি জানো না কারা ধনী আর কারা কিপটে। মডেলকে দিলে সে তাদের খরচের ধরন দেখে ৩-৪টি গ্রুপে ভাগ করে দেবে।
    - অ্যানোমালি ডিটেকশন **(Anomaly Detection):** ক্রেডিট কার্ডের ট্রানজ্যাকশনে হঠাৎ কোনো অস্বাভাবিক প্যাটার্ন খুঁজে বের করা।
    - টপিক মডেলিং: হাজার হাজার নিউজ আর্টিকেল থেকে অটোমেটিক বের করা যে কোনটা স্পোর্টস আর কোনটা পলিটিক্স।

২. **Reinforcement Learning** কখন ব্যবহার করবে?

যখন কোনো একটি কাজ করার জন্য বারবার চেষ্টা ও ভুল **(Trial and Error)** করার সুযোগ থাকে এবং প্রতিটি কাজের পর একটি পুরস্কার বা শাস্তি (Reward/Penalty) পাওয়া যায়। এখানে এজেন্টকে সিদ্ধান্ত নিতে হয়।

- উদ্দেশ্য: সময়ের সাথে সাথে সবচাইতে বেশি পুরস্কার (Reward) পাওয়ার কৌশল বা 'Policy' শেখা।
- কখন ব্যবহার করবে:
    - গেম খেলা **(Gaming):** যেখানে প্রতিটি চালের পর জয় বা পরাজয়ের সম্ভাবনা থাকে (যেমন: দাবা, লুডু বা ভিডিও গেম)।
    - রোবোটিক্স: একটি রোবটকে হাঁটতে শেখানো। সে যখন পড়ে যায় (Penalty), তখন সে শেখে যে ওইভাবে পা ফেলা যাবে না।
    - অটোনোমাস কার: গাড়ি চালানো শেখা। সিগন্যাল মানলে রিওয়ার্ড, আর ধাক্কা লাগলে পেনাল্টি।
    - ইনভেন্টরি ম্যানেজমেন্ট: কখন কতটুকু মাল অর্ডার করলে লাভ সবচাইতে বেশি হবে এবং লস কম হবে।

৩. এদের মধ্যে মূল পার্থক্য (তুলনা)

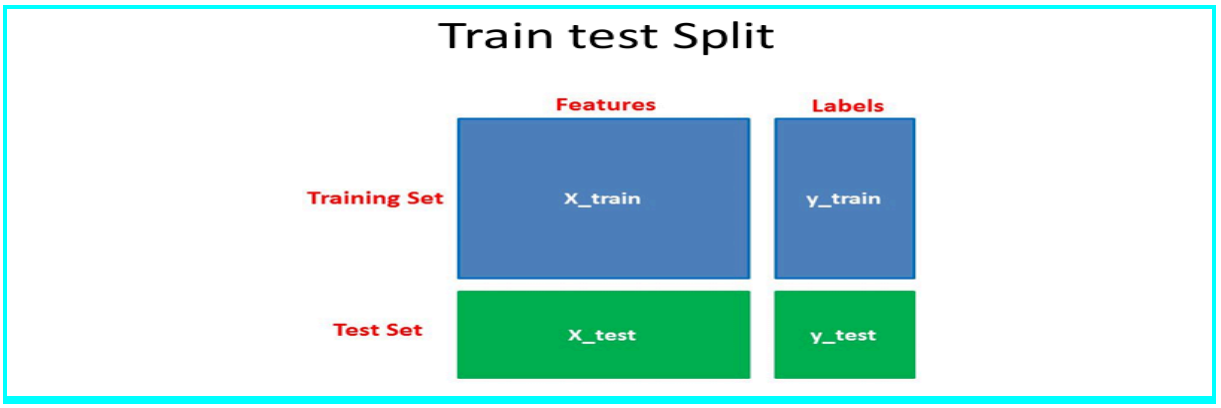
বৈশিষ্ট্য	**Unsupervised Learning**	**Reinforcement Learning**
মূল বিষয়	ডেটার ভেতরে থাকা মিল/প্যাটার্ন খুঁজে বের করা।	কাজের মাধ্যমে সেরা উপায় **(Strategy)** শেখা।
ফিডব্যাক	কোনো ফিডব্যাক বা রিওয়ার্ড নেই।	রিওয়ার্ড বা পেনাল্টি (পুরস্কার বা শাস্তি) আছে।
পরিবেশ	স্ট্যাটিক বা স্থির ডেটাসেট নিয়ে কাজ করে।	ডায়নামিক বা পরিবর্তনশীল পরিবেশে কাজ করে।
সিদ্ধান্ত	এখানে কোনো সিদ্ধান্ত বা 'অ্যাকশন' নেওয়ার ব্যাপার নেই।	এখানে প্রতি ধাপে একটি 'অ্যাকশন' নিতে হয়।

৪. একটি বাস্তব উদাহরণ দিয়ে বোঝো:

মনে করো তুমি একটি শপিং মল চালাচ্ছো।

- **Unsupervised:** তুমি দেখলে কোন বয়সের মানুষ কোন সময় শপিং করতে আসে এবং তাদের আলাদা আলাদা গ্রুপে ভাগ করলে। এটি তোমাকে কাস্টমারদের বুঝতে সাহায্য করলো।
- **Reinforcement:** তুমি ঠিক করলে কাস্টমারদের ডিসকাউন্ট কুপন দেবে। তুমি বিভিন্ন সময়ে বিভিন্ন কুপন দিয়ে দেখলে কোন কুপন দিলে মানুষ বেশি কেনাকাটা করছে (Reward)। এভাবে ট্রায়াল অ্যান্ড এরর এর মাধ্যমে তুমি সেরা ডিসকাউন্ট অফারটি খুঁজে বের করলে।

সারসংক্ষেপ: আপনার কাছে শুধু স্থির কিছু ডেটা থাকলে এবং সেখান থেকে গ্রুপ বের করতে চাইলে **Unsupervised** ব্যবহার করুন। আর যদি এমন কোনো সিচুয়েশন হয় যেখানে মেশিনকে নিজে থেকে কোনো কাজ করতে হবে এবং ভুল থেকে শিখতে হবে, তবে **Reinforcement Learning** ব্যবহার করুন।



১. Train-Test Split

আমরা যখন কোনো মডেল বানাই, তখন আমাদের কাছে থাকা সম্পূর্ণ ডেটাসেটকে আমরা দুই ভাগে ভাগ করি। কেন? কারণ মডেলটি কতটা ভালো শিখল তা পরীক্ষা করার জন্য আমাদের কাছে কিছু "অচেনা" ডেটা থাকতে হবে।

- **Training Set** (নীল অংশ)**:** এটি হলো সেই ডেটা যা দিয়ে আমরা মডেলকে পড়াশোনা করাই।
  - **X_train:** এখানে সব ইনপুট কলাম বা Features থাকে।
  - **y_train:** এখানে সঠিক উত্তর বা Labels থাকে।
- **Test Set** (সবুজ অংশ)**:** এটি মডেলের জন্য "পরীক্ষার প্রশ্নপত্র"। মডেলটি আগে কখনো এই ডেটা দেখেনি।
  - **X_test:** মডেলকে শুধু ইনপুটগুলো দেওয়া হয় এবং তাকে প্রেডিক্ট করতে বলা হয়।
  - **y_test:** এটি হলো আসল উত্তর। মডেলের প্রেডিকশনের সাথে এই y_test মিলিয়ে দেখা হয় মডেলটি কতটা নির্ভুল।

১. High Bias (হাই বায়াস) — Underfitting-এর সাথে যুক্ত

Bias মানে হলো মডেলটি ডেটা সম্পর্কে কতটা ভুল ধারণা বা "সরল বিশ্বাস" নিয়ে বসে আছে।

- কী বোঝানো হয়**:** মডেলটি যখন খুব বেশি সহজ (Simple) হয়, তখন সে ডেটার জটিল প্যাটার্নগুলো ধরতেই পারে না। সে ধরে নেয় ডেটা খুব সাধারণ, কিন্তু আসলে তা নয়।
- ফলাফল**:** এটি ট্রেনিং ডেটাতেও ভুল করে, আবার টেস্ট ডেটাতেও ভুল করে।
- ছবিটির সাথে কানেকশন**:** একেই আমরা ছবিতে **Underfitting** হিসেবে দেখেছি।

২. High Variance (হাই ভ্যারিয়েন্স) — Overfitting-এর সাথে যুক্ত

Variance মানে হলো মডেলটি ডেটার সামান্য পরিবর্তনেই কতটা "উত্তেজিত" হয়ে পড়ে বা দিক পরিবর্তন করে।

- কী বোঝানো হয়: মডেলটি যখন খুব বেশি জটিল (Complex) হয়, তখন সে ট্রেনিং ডেটার প্রতিটি ছোট ছোট পয়েন্ট এমনকি 'নয়েজ' বা ভুলগুলোকেও মুখস্থ করে ফেলে। সে শুধু ওই নির্দিষ্ট ডেটাসেটেই ভালো রেজাল্ট দেয়।
- ফলাফল: ট্রেনিং ডেটাতে পারফরম্যান্স চমৎকার (Low error), কিন্তু নতুন কোনো ডাটা (Test data) দিলেই সে খেই হারিয়ে ফেলে এবং ভুল রেজাল্ট দেয়।
- ছবিটির সাথে কানেকশন: একেই আমরা ছবিতে **Overfitting** হিসেবে দেখেছি।

৩. Bullseye (লক্ষ্যভেদ) উদাহরণ দিয়ে বোঝো:

কল্পনা করো তুমি একটি তিরে লক্ষ্যভেদ করছো (Target hitting):

1. **Low Bias, Low Variance (নিখুঁত):** সবগুলো তির একদম মাঝখানে (Target) লাগছে। এটাই আমাদের লক্ষ্য।
2. **High Bias, Low Variance:** তিরগুলো মাঝখানে লাগছে না, বরং এক কোণায় স্তূপ হয়ে আছে। অর্থাৎ মডেলটি ভুল জায়গায় স্থির হয়ে আছে (মডেলটি খুব সহজ)।
3. **Low Bias, High Variance:** তিরগুলো লক্ষ্যের আশেপাশে আছে কিন্তু খুব ছড়িয়ে ছিটিয়ে। অর্থাৎ মডেলটি একেক ডেটায় একেক রকম আচরণ করছে (মডেলটি ডেটা মুখস্থ করেছে)।
4. **High Bias, High Variance:** তিরগুলো মাঝখান থেকেও দূরে এবং মারাত্মকভাবে ছড়ানো। এটা সবচাইতে খারাপ অবস্থা।

---

৪. সারাংশ টেবিল:

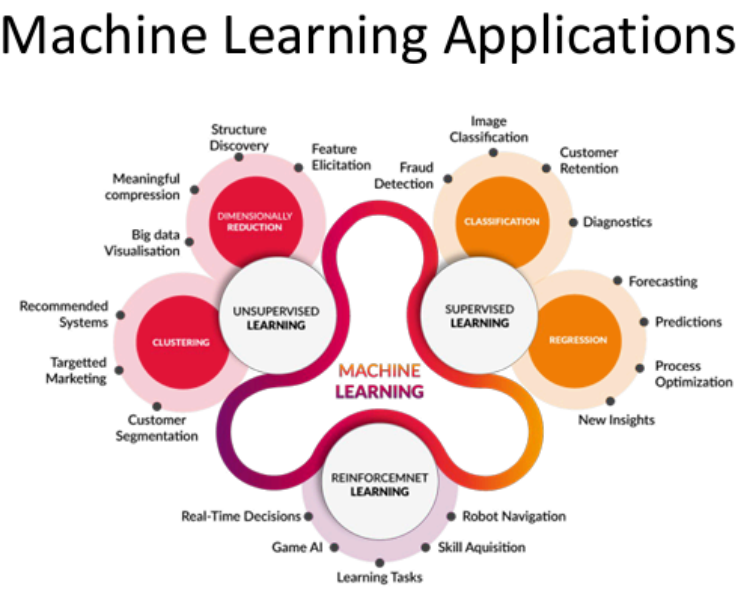
টার্ম	যুক্ত আছে যার সাথে	মডেলের অবস্থা	ট্রেনিং এরর	টেস্ট এরর
High Bias	Underfitting	খুব সিম্পল	High	High
High Variance	Overfitting	খুব কমপ্লেক্স	Low	High

Bias-Variance Tradeoff:

মেশিন লার্নিংয়ের মূল চ্যালেঞ্জ হলো বায়াস কমানো এবং ভ্যারিয়েন্স কমানোর মধ্যে একটি ব্যালেন্স বা ভারসাম্য তৈরি করা। আমরা এমন এক জায়গায় পৌঁছাতে চাই যেখানে বায়াস এবং ভ্যারিয়েন্স—দুটোই সর্বনিম্ন থাকে।

বৈশিষ্ট্য	**Classification**	**Clustering**
শিক্ষার ধরণ	Supervised (শিক্ষক বা লেবেল আছে)	Unsupervised (শিক্ষক বা লেবেল নেই)

আউটপুট	আগে থেকে নির্ধারিত (Pre-defined) ক্লাসে ভাগ করা।	নতুন কোনো গ্রুপ বা প্যাটার্ন তৈরি করা।
লক্ষ্য	নতুন ডাটা কোন ক্লাসের তা প্রেডিক্ট করা।	ডাটার মধ্যে লুকানো মিল খুঁজে বের করা।
প্রক্রিয়া	ট্রেনিং এবং টেস্টিং—এই দুই ধাপে হয়।	শুধু সরাসরি ডাটা প্রসেসিং হয়।
উদাহরণ	Decision Tree, SVM, Naive Bayes.	K-Means, Hierarchical Clustering.



APPLICATIONS:

১. Supervised Learning (তত্ত্বাবধানে শিক্ষা)

এখানে আমরা জানি আমরা কী খুঁজছি। এর দুটি প্রধান শাখা ছবিতে দেখানো হয়েছে:

A. Classification (শ্রেণিবিন্যাস)

- **Fraud Detection:** ক্রেডিট কার্ডের লেনদেন কি আসল নাকি জালিয়াত?
- **Image Classification:** একটি ছবিতে কী আছে তা চেনা (যেমন: ফেসবুকের ফেস ডিটেকশন)।

- **Customer Retention:** কোন কাস্টমার আমাদের সার্ভিস ছেড়ে চলে যেতে পারে তা আগে থেকে বোঝা।
- **Diagnostics:** ডাটা দেখে রোগ নির্ণয় করা (যেমন: ক্যানসার ডিটেকশন)।

B. Regression (ভবিষ্যদ্বাণী)

- **Forecasting/Predictions:** শেয়ার বাজারের দাম বা আগামীকালের তাপমাত্রা কত হবে তা বলা।
- **Process Optimization:** কোনো উৎপাদন প্রক্রিয়ায় ইনপুট কতটুকু দিলে আউটপুট সেরা হবে তা বের করা।
- **New Insights:** ডাটার মধ্যকার গাণিতিক সম্পর্ক খুঁজে বের করা।

---

২. Unsupervised Learning (স্বয়ংক্রিয় শিক্ষা)

এখানে ডাটার কোনো লেবেল থাকে না। মেশিন নিজেই মিল খুঁজে বের করে। এর শাখাগুলো হলো:

A. Clustering (গুচ্ছকরণ)

- **Recommended Systems:** তুমি নেটফ্লিক্সে একটি মুভি দেখলে তোমাকে একই রকম আরও মুভি সাজেস্ট করা।
- **Targeted Marketing:** কেনাকাটার ধরন দেখে কাস্টমারদের গ্রুপ করা।
- **Customer Segmentation:** একই বৈশিষ্ট্যের ক্রেতাদের আলাদা করা।

B. Dimensionality Reduction (আকার কমানো)

- **Structure Discovery:** ডাটার ভেতরের মূল কাঠামো খুঁজে বের করা।
- **Meaningful Compression:** অপ্রয়োজনীয় তথ্য বাদ দিয়ে ডাটাকে ছোট করা যাতে প্রসেসিং সহজ হয়।
- **Big Data Visualization:** কোটি কোটি ডাটাকে ছোট করে গ্রাফে সুন্দরভাবে দেখানো।

---

৩. Reinforcement Learning (অভিজ্ঞতা থেকে শিক্ষা)

এটি মূলত 'Trial and Error' বা পুরস্কার ও শাস্তির মাধ্যমে কাজ করে। এর প্রয়োগগুলো হলো:

- **Game AI:** দাবা বা ভিডিও গেমে কম্পিউটার যখন মানুষের সাথে খেলে (যেমন: AlphaGo)।
- **Robot Navigation:** একটি রোবট বাধা এড়িয়ে কীভাবে এক জায়গা থেকে অন্য জায়গায় যাবে।
- **Real-time Decisions:** মুহূর্তের মধ্যে সিদ্ধান্ত নেওয়া (যেমন: ড্রাইভলেস কারের ব্রেক কষার সিদ্ধান্ত)।
- **Skill Acquisition:** রোবট বা সফটওয়্যারকে নতুন কোনো দক্ষতা (যেমন: হাত দিয়ে কিছু ধরা) শেখানো।

সারসংক্ষেপ:

- যদি তোমার কাছে অতীতের উত্তরসহ ডাটা থাকে → **Supervised** (প্রেডিকশন বা ক্লাসিফিকেশন করার জন্য)।
- যদি তোমার কাছে শুধু বিশাল স্তূপ ডাটা থাকে এবং তুমি প্যাটার্ন দেখতে চাও → **Unsupervised** (গ্রুপ করার জন্য)।
- যদি মেশিনকে ভুল থেকে শিখিয়ে কোনো কাজ করাতে চাও → **Reinforcement** (রোবট বা গেমের জন্য)।

এই অ্যাপ্লিকেশানগুলোর মধ্যে কোনোটি কি তোমার প্রজেক্ট বা পড়াশোনার জন্য বিশেষভাবে দরকার? তাহলে আমি সেটি নিয়ে আরও বিস্তারিত উদাহরণ দিতে পারি।